Plant Monitoring

SUNNY WEBBOX RPC

User Manual

Remote Procedure Call Description
Interfaces and API Definition

Legal Provisions

Copyright © 2013 SMA America, LLC. All rights reserved.

No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, be it electronic, mechanical, photographic, magnetic or otherwise, without the prior written permission of SMA America, LLC.

Neither SMA America, LLC nor SMA Solar Technology Canada Inc. makes representations, express or implied, with respect to this documentation or any of the equipment and/or software it may describe, including (with no limitation) any implied warranties of utility, merchantability, or fitness for any particular purpose. All such warranties are expressly disclaimed. Neither SMA America, LLC nor its distributors or dealers nor SMA Solar Technology Canada Inc. nor its distributors or dealers shall be liable for any indirect, incidental, or consequential damages under any circumstances.

(The exclusion of implied warranties may not apply in all cases under some statutes, and thus the above exclusion may not apply.)

Specifications are subject to change without notice. Every attempt has been made to make this document complete, accurate and up-to-date. Readers are cautioned, however, that SMA America, LLC and SMA Solar Technology Canada Inc. reserve the right to make changes without notice and shall not be responsible for any damages, including indirect, incidental or consequential damages, caused by reliance on the material presented, including, but not limited to, omissions, typographical errors, arithmetical errors or listing errors in the content material.

All trademarks are recognized even if these are not marked separately. Missing designations do not mean that a product or brand is not a registered trademark.

The *Bluetooth*® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by SMA America, LLC and SMA Solar Technology Canada Inc. is under license.

SMA America, LLC

3801 N. Havana Street

Denver, CO 80239 U.S.A.

SMA Solar Technology Canada Inc.

2425 Matheson Blvd. E

7th Floor

Mississauga, ON L4W 5K4

Canada

Important Safety Instructions

SAVE THESE INSTRUCTIONS

This manual contains important instructions for the following products:

- Sunny WebBox
- Sunny WebBox with *Bluetooth*® Wireless Technology

This manual must be followed during installation and maintenance.

The product is designed and tested in accordance with international safety requirements, but as with all electrical and electronic equipment, certain precautions must be observed when installing and/or operating the product. To reduce the risk of personal injury and to ensure the safe installation and operation of the product, you must carefully read and follow all instructions, cautions and warnings in this manual.

Warnings in this document

A warning describes a hazard to equipment or personnel. It calls attention to a procedure or practice, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the SMA equipment and/or other equipment connected to the SMA equipment or personal injury.

Symbol	Description
⚠ DANGER	DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.
⚠ WARNING	WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.
⚠ CAUTION	CAUTION indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.
NOTICE	NOTICE is used to address practices not related to personal injury.

General Warnings

⚠ WARNING

General Warnings

All electrical installations must be made in accordance with the local and *National Electrical Code*® ANSI/NFPA 70 or the *Canadian Electrical Code*® CSA C22.1. This document does not and is not intended to replace any local, state, provincial, federal or national laws, regulation or codes applicable to the installation and use of the product, including without limitation applicable electrical safety codes. All installations must conform with the laws, regulations, codes and standards applicable in the jurisdiction of installation. SMA assumes no responsibility for the compliance or noncompliance with such laws or codes in connection with the installation of the product.

The product contains no user-serviceable parts.
For all repair and maintenance, always return the unit to an authorized SMA Service Center.

Before installing or using the product, read all of the instructions, cautions, and warnings in this manual.

Wiring of the product must be made by qualified personnel only.

Table of Contents

1 Introduction 9
1.1 Validity 9
1.2 Nomenclature 9
1.3 Acronyms and Abbreviations 9
1.4 Referenced Documents and Sources 10
2 System Overview 11
2.1 Sunny WebBox RPC 11
2.2 Differences between Sunny WebBox and Sunny WebBox with *Bluetooth* *11*
3 JavaScript Object Notation 12
3.1 Example 12
4 Procedure Conventions 14
4.1 Procedure Call (Request) 14
4.2 Return Value (Response) 15
4.2.1 JSON 15
4.3 Query Interval 15
5 Interfaces 16
5.1 RPC via UDP Stream 16
5.2 RPC via HTTP 16
6 Object Definitions 17
6.1 Device Object 17
6.2 Channel Object 18
7 Service Procedures 19
7.1 RPC_GET_PLANT_OVERVIEW 19
7.1.1 Version 1.0 19
7.2 RPC_GET_DEVICES 21
7.2.1 Version 1.0 21
7.3 RPC_GET_PROCESS_DATA_CHANNELS 23

7.3.1 Version 1.0. 23
7.4 RPC_GET_PROCESS_DATA 25
7.4.1 Version 1.0. 25
7.5 RPC_GET_PARAMETER_CHANNELS. 28
7.5.1 Version 1.0. 28
7.6 RPC_GET_PARAMETER. 30
7.6.1 Version 1.0. 30
7.7 RPC_SET_PARAMETER 35
7.7.1 Version 1.0. 35
8 Compliance Information 39
9 Contact 40

1 Introduction

1.1 Validity

This document applies to the RPC interface of the Sunny WebBox and the Sunny WebBox with *Bluetooth*® Wireless Technology.

1.2 Nomenclature

Complete designation	Designation in this document
SMA America Production, LLC	SMA
SMA Solar Technology Canada Inc.	SMA
Sunny WebBox with *Bluetooth*® Wireless Technology	Sunny WebBox with *Bluetooth*

1.3 Acronyms and Abbreviations

API	Application Programmers Interface
HTTP	Hypertext Transfer Protocol
JSON	Javascript Object Notation
UDP	User Data Protocol
URL	Uniform Resource Locator
RAS	Remote Access Service
RFC	Request for Comment
RPC	Remote Procedure Call

1.4 Referenced Documents and Sources

[1]
RFC 4627: The application/json Media Type for Javascript Object Notation (JSON)
http://www.ietf.org/rfc/rfc4627.txt?number=4627

[2]
JSON-RPC 1.1 Working Draft August 2006
http://json-rpc.org/wd/JSON-RPC-1-1-WD-20060807.html

[3]
Introducing JSON
http://www.json.org

[4]
The MD5 Message-Digest Algorithm
http://www.ietf.org/rfc/rfc1321.txt?number=1321

2 System Overview

2.1 Sunny WebBox RPC

The data loggers Sunny WebBox and Sunny WebBox with *Bluetooth* continuously record all the data of a PV plant. This is then averaged over a configurable interval and cached. The data can be transmitted at regular intervals to the Sunny Portal for analysis and visualization.

Via the Sunny WebBox and Sunny WebBox with *Bluetooth* RPC interface, selected data from the PV plant can be transmitted to a remote terminal by means of an RPC protocol (Remote Procedure Call protocol).

To do this, the Sunny WebBox provides a pool of service procedures (see Section 6 "Object Definitions", page 17) that can be accessed from the remote terminal by means of a Remote Procedure Call protocol (RPC protocol) via a network or RAS connection. The data exchange format used here is the JavaScript Object Notation (JSON) (see Section 3 "JavaScript Object Notation", page 12).

2.2 Differences between Sunny WebBox and Sunny WebBox with *Bluetooth*

Different representation for Sunny WebBox and Sunny WebBox with *Bluetooth*

The presentation of the plant data transmitted via the RPC interface is different for the Sunny WebBox and the Sunny WebBox with *Bluetooth*.

This document only contains sample responses for the Sunny WebBox and none from the Sunny WebBox with *Bluetooth*.

You will find a list of measured values and parameters for the Sunny WebBox with *Bluetooth* in the technical description "Measured Values and Parameters" at www.SMA-Solar.com.

Available service procedures	Sunny WebBox	Sunny WebBox with *Bluetooth*
RPC_GET_PLANT_OVERVIEW	+	+
RPC_GET_DEVICES	+	+
RPC_GET_PROCESS_DATA_CHANNELS	+	+
RPC_GET_PROCESS_DATA	+	+
RPC_GET_PARAMETER_CHANNELS	+	–
RPC_GET_PARAMETER	+	–
RPC_SET_PARAMETER	+	–

3 JavaScript Object Notation

A description and explanation of JavaScript Object Notation (JSON) can be found on the website http://www.json.org.

3.1 Example

The following example shows an illustration of a device list. It defines an object made up of the values "totalDevicesReturned" and "devices".

"totalDevicesReturned" is a figure with the value 4. The array "device" has 2 fields, each having one device object (see Section 6.1 "Device Object", page 17), which in turn contains nested device objects.

```
RPC={
    "totalDevicesReturned":4,
    "devices":
    [
     {
      "key":"SCC250H9:1390148531",
      "name":"Sunny Central E1",
      "children":
      [
       {
        "key":"SCBFS016:8945",
        "name":"Sunny BFS E1",
        "children":null
       },
       {
        "key":"SMU8b004:2567",
        "name":"String Monitoring Unit E1",
        "children":null
       }
      ]
     },
     {
      "key":"SCC250H9:1390148538",
      "name":"Sunny Central E2",
      "children":
```

```
      [

        {
          "key":"SCBFS016:8956",
          "name":"Sunny BFS E2",
          "children":null
        },
        {
          "key":"SMU8b004:2534",
          "name":"String Monitoring Unit E2",
          "children":null
        }
      ]
    }
  ]
}
```

4 Procedure Conventions

All identifiers used are case-sensitive. This means that "Power" and "power", for instance, designate two different objects. All characters are transmitted using Unicode in UTF-8 format.

4.1 Procedure Call (Request)

Each request consists of one serialized JSON object which possesses the following obligatory members:

- **version** – a character sequence which defines the underlying RPC version.
- **proc** – a character sequence which contains the procedure to be called.
- **id** – a random character sequence (max. 16 characters) which serves to assign a response to the request.
- **format** – a character sequence defining the data exchange format of the procedure's result (see Section 4.2 "Return Value (Response)", page 15).
- **passwd** – a character sequence comprising the hash value of the password for the desired access level (user, installer). The hash value is calculated using the MD5 algorithm (see [4]). Password allocation is performed via the security settings in the Sunny WebBox. If the object is not transmitted, the user level is automatically assumed.
- **params** – an object whose elements are transferred to the procedure as arguments. Each parameter must be available in the form of a named JSON object. Hence, the sequence is arbitrary. The number of parameters depends on the given service procedure (see Section 7). If the requested procedure does not expect any arguments, the entry is omitted.

4.2 Return Value (Response)

The data exchange format of the response is defined by the character sequence transmitted in the request.

The following formats are currently available:

- JSON

4.2.1 JSON

The response to a request consists of a serialized JSON object made up of the following compulsory objects:

- **version** – a character sequence which defines the underlying RPC version.
- **proc** – a character sequence which contains the called procedure.
- **id** – a character sequence which serves to assign the request and response. Contains the ID from the respective request.
- **result** – the result of the executed procedure as a serialized JSON object. If due to an error the procedure cannot be executed successfully, the "error" object will be transmitted instead.
- **error** – an object containing a character string with a description of the error encountered. If the procedure is executed without errors, this object will not be transmitted.

4.3 Query Interval

The interval between two queries should not be less than 30 seconds.

5 Interfaces

The Sunny WebBox provides two different access options which differ in the implementation effort required and their runtime utilization of resources.

5.1 RPC via UDP Stream

The procedure call is transmitted to port 34268 of the Sunny WebBox in the usable data part of the UDP protocol. Responses are also sent to port 34268.

UDP transport requires a relatively low implementation effort on the client side and saves runtime resources. For communication beyond the limits of local networks, the port will normally need to be opened by the appropriate firewalls.

5.2 RPC via HTTP

Data exchange takes place by means of the Hypertext Transfer Protocol via a TCP/IP connection to the web server port configured in the Sunny WebBox.

The default setting is port 80.

The URL for all requests is: http://IP address/rpc

The IP address in each case is the currently configured IP address of the Sunny WebBox. The default setting is 192.168.0.168.

Hence, the default URL is the following: http://192.168.0.168/rpc

The request is transmitted via HTTP POST in the body of the HTTP request as a serialized JSON object according to the conventions established in Section 4.1.

Both the client-side implementation effort and the resource requirements are relatively high. As a general rule, communication takes place via the standard port 80, which means that there is no need to make any changes to active firewalls.

6 Object Definitions

This section defines the structure of frequently used objects using the JSON syntax. In the description, the values of the object members define the type of data used. All definitions apply accordingly to all other data transmission formats.

6.1 Device Object

Describes a device within the plant (e.g. Sunny Boy, Sunny Sensor Box).

RPC={

"key": "string",

"name": "string" or null (optional),

"channels": [array] or null (optional),

"children": [array] or null (optional)

}

key: A unique device key (e.g. "SB21TL06:2000106925").

name: The user-definable name of the device (e.g. "INV left"). Defining this element is optional. If the element is used but no name was defined, "null" is entered.

channels: An array of channel objects of the device. Defining this element is optional. If the element is used but no channels exist, "null" is entered.

children: An array of objects containing sub-devices. Defining this element is optional. If the element is used but no sub-devices exist, "null" is entered.

6.2 Channel Object

Describes a process data or parameter channel of a device.

```
RPC={
        "meta": "string",
        "name": "string" (optional),
        "value": "string",
        "unit": "string" (optional),
        "min": "string" (optional),
        "max": "string" (optional),
        "options": [array] (optional)
      }
```

meta:	The meta name which uniquely defines the channel (e.g. "ExtSolIrr").
name:	The translated display name (e.g. "External irradiation"). Defining this element is optional.
value:	The value of the channel (e.g. "843"). Defining this element is optional.
unit:	The unit of the channel (e.g. "W/m^2). For channels which have no unit, an empty string must be entered.
min:	The minimum possible value of a channel. Defining this element is optional.
max:	The maximum possible value of a channel. Defining this element is optional.
options:	A list with the possible values of a parameter channel. Defining this element is optional.

7 Service Procedures

This section describes the structure of the available service procedures.

For each procedure, a brief description of its task is given. This is followed by the structure of the request. Variable elements are represented by placeholders in upper-case characters. In this manual, the placeholders VERSION, FORMAT, and ID are not described for each procedure, since their meaning has already been described in Section 4 and it is not necessary to differentiate between all procedures here.

7.1 RPC_GET_PLANT_OVERVIEW

7.1.1 Version 1.0

Returns an object with the following plant data:

- POWER
- DAILY-YIELD
- TOTAL-YIELD
- STATUS
- ERROR

Structure:

```
RPC={
      "version": "1.0",
      "proc": "GetPlantOverview",
      "id": "ID",
      "format": "FORMAT"
     }
```

Sample request:

```
RPC={
      "version": "1.0",
      "proc": "GetPlantOverview",
      "id": "1",
      "format": "JSON"
     }
```

Sample response:


```
RPC={
        "version": "1.0",
        "proc": "GetPlantOverview",
        "id": "1",
        "result":
        {
          "overview":
          [
            {
              "meta": "GriPwr",
              "name": "Momentanleistung",
              "value": "4250",
              "unit": "W"
            },
            {
              "meta": "GriEgyTdy",
              "name": "Tagesenergie",
              "value": "45.23",
              "unit": "kWh"
            },
            {
              "meta": "GriEgyTot",
              "name": "Gesamtenergie",
              "value": "7821",
              "unit": "kWh"
            },
            {
              "meta": "OpStt",
              "name": "Status",
              "value": "MPP",
              "unit": null
            },
            {
              "meta": "Msg",
```

```
            "name": "Fehler",
            "value": null,
            "unit": null
          }
        ]
      }
    }
```


The following data was transmitted:

Power = 4,250 W,
E-Today = 45.23 kWh,
E-Total = 7,821 kWh,
Status = MPP,
No error

7.2 RPC_GET_DEVICES

7.2.1 Version 1.0

Returns a hierarchical list of all detected plant devices.

Structure:


```
RPC={
      "version": "1.0",
      "proc": "GetDevices",
      "id": "ID",
      "format": "FORMAT"
    }
```


Sample request:


```
RPC={
      "version": "1.0",
      "proc":   "GetDevices",
      "id": "1",
      "format": "JSON"
    }
```

Sample response:


```
RPC={
    "version": "1.0",
    "proc": "GetDevices",
    "id": "1",
    "result":
    {
     "totalDevicesReturned": 6,
     "devices":
     [
      {
       "key":"SCC250H9: 1390148531",
       "name:" Sunny Central E1",
       "children":
       [
        {
         "key": "SCBFS016:8945",
         "name": "Sunny BFS E1",
         "children": null
        },
        {
         "key": "SMU8b004:2567",
         "name": "String Monitoring Unit E1",
         "children": null
        }
       ]
      },
      {
       "key": "SCC250H9:1390148538",
       "name": "Sunny Central E2",
       "children":
       [
        {
         "key": "SCBFS016:8956",
         "name": "Sunny BFS E2",
         "children": null
```

```
        },
        {
          "key": "SMU8b004:2534",
          "name": "String Monitoring Unit E2",
          "children": null
        }
       ]
      }
     ]
    }
   }
```

7.3 RPC_GET_PROCESS_DATA_CHANNELS

7.3.1 Version 1.0

Returns a list with the meta names of the available process data channels for a particular device type.

Structure:

```
RPC={
    "version": "1.0",
    "proc": "GetProcessDataChannels",
    "id": "ID",
    "format": "FORMAT",
    "params":
    {
      DEVICE_KEY
    }
   }
```

DEVICE_KEY: The device key of a device of the type for which the process data channels are to be returned.

Sample request:

```
RPC={
        "version": "1.0",
        "proc": "GetProDataChannels",
        "id": "1",
        "format": "JSON",
        "params":
        {
          "device": "WR715-19:263415747"
        }
       }
```

Sample response:

```
RPC={
        "version": "1.0",
        "proc": "GetProcessDataChannels",
        "id": "1",
        "result":
        {
          "WR715-19:263415747":
          [
            "Upv-Soll",
            "h-Total",
            "Zac",
            "Status",
            "E-Total",
            "Upv-Ist",
            "Riso",
            "Uac",
            "Pac",
            "Fehler-Cnt",
            "Ipv",
            "Netz-Ein",
            "Seriennummer",
            "Fac"
            "Fehler"
```

```
        "lac-lst"
      ]
    }
  }
```

7.4 RPC_GET_PROCESS_DATA

7.4.1 Version 1.0

Returns process data for up to 5 devices per request.

Structure:

```
RPC={
    "version": "1.0",
    "proc": "GetProcessData",
    "id": "ID",
    "format": "FORMAT",
    "params":
    {
      "DEVICES":
      [
        {
          "key": DEVICE_KEY,
          "channels": [CHANNELS]
        }
      ]
    }
  }
```

A list with the device keys whose process data is to be provided must be transferred as a parameter. You can submit a selection of required process data to each device. If a selection is omitted, all process data will be transmitted.

DEVICES:	An array containing objects with the device keys of those devices for which process data is to be returned, and optional CHANNELS.
DEVICE_KEY:	The corresponding device key (see Section 6.1 "Device Object", page 17).
CHANNELS:	An array containing the meta names of the required process data. The available meta names can be identified with the command RPC_GET_PROCESS_DATA_CHANNELS.

Sample request:

```
RPC={
        "version": "1.0",
        "proc": "GetProcessData",
        "id": "1",
        "format": "JSON",
        "params":
        {
          "devices":
          [
            {
              "key": "WR715-19:263415747",
              "channels": null
            },
            {
                              "key": "WR715-19:263415748","key": "WR715-19:263415748",
              "channels":
              [
                "Pac"
              ]
            }
          ]
        }
      }
```

Sample response:

```
RPC={
        "version": "1.0",
        "proc": "GetProcessData",
        "id": "1",
        "result":
        {
          "devices":
          [
            {
              "key": "WR715-19:263415747",
              "channels":
```

```
        [
          {
            "meta": "E-Total",
            "name": null,
            "value": "1160.987",
            "unit": "kWh"
          },
          {
            "meta": "Fac",
            "name": null,
            "value": "49.98",
            "unit": "Hz"
          },
          {
            "meta": "Zac",
            "name": null,
            "value": "1.346",
            "unit": "Ohm"
          }
        ]
      },
      {
                  "key": "WR715-19:263415748","key": "WR715-19:263415748",
        "channels":
        [
          {
            "meta": "Pac",
            "name": null,
            "value": "630",
            "unit": "W"
          }
        ]
      }
    ]
  }
}
```

7.5 RPC_GET_PARAMETER_CHANNELS

7.5.1 Version 1.0

Returns a list with the meta names of the available parameter channels for a particular device type, depending on access level. The level is determined by transmitting the MD5 hash value of the respective password in the request header.

Structure:

```
RPC={
      "version": "1.0",
      "proc": "GetParameterChannels",
      "id": "ID",
      "format": "FORMAT",
      "passwd" : "PASSWORD",
      "params":
      {
        "key": DEVICE_KEY
      }
    }
```

PASSWORD: The MD5 coded hash value of the password for the desired access level.

DEVICE_KEY: The device key of a device for whose type the parameter channels are to be returned.

Sample request

```
RPC={
      "version": "1.0",
      "proc": "GetParameterChannels",
      "id": "1",
      "format": "JSON",
      "passwd": "a289fa4252ed5af8e3e9f9bee545c172",
      "params":
      {
        "device": "WR715-19:263415747"
      }
    }
```

Sample response:


```
RPC={
        "version": "1.0",
        "proc": "GetParameterChannels",
        "id": "1",
        "result":
        {
          "WR715-19:263415747":
          [
            "Plimit",
            "SMA-Grid-Guard",
            "SMA-SN",
            "Betriebsart",
            "Control",
            "Ripple-Ctl-Frq",
            "PowerBalancer",
            "Usoll-Konst",
            "Upv-Start",
            "Default",
            "T-Start",
            "Ripple-Ctl-Lev",
            "Storage",
            "Ripple-Ctl-Rcvr",
            "Software-SRR",
            "T-Stop",
            "Software-BFR",
            "Hardware-BFS"
          ]
        }
      }
```

7.6 RPC_GET_PARAMETER

7.6.1 Version 1.0

Returns the parameter values of up to 5 devices, depending on the access level. The level is determined by transmitting the MD5 hash value of the respective password in the request header.

Structure:

```
RPC={
    "version": "1.0",
    "proc": "GetParameter",
    "id": "ID",
    "format": "FORMAT",
    "passwd": "PASSWORT",
    "params":
    {
      "DEVICES":
      [
        {
          "key": DEVICE_KEY,
          "channels": [CHANNELS]
        }
      ]
    }
  }
```

A list with the device objects whose parameters are to be provided must be transferred as a parameter. You can submit a selection of requested parameters to each device. If this selection is omitted, all parameters will be transmitted.

PASSWORD:	The MD5 coded hash value of the password for the desired access level.
DEVICES:	An array containing the device objects for which parameters are to be returned, and an optional selection of certain channels.
DEVICE_KEY:	The corresponding device key (see Section 6.1 "Device Object", page 17).
CHANNELS:	An array containing the meta names of the required process data. The available meta names can be identified with the command RPC_GET_PROCESS_DATA_CHANNELS.

Sample request

```
RPC={
        "version": "1.0",
        "proc": "GetParameter",
        "id": "1",
        "format": "JSON",
        "passwd": "a289fa4252ed5af8e3e9f9bee545c172",
        "params":
        {
          "devices":
          [
            {
              "key": "WR715-19:263415747"
            }
          ]
        }
      }
```

Sample response:

```
RPC={
        "version": "1.0",
        "id": "1",
        "format": "JSON",
        "proc": "GetParameter",
        "result":
        {
          "devices":
          [
            {
              "key": "WR21TL06:2000101000"
              "channels":
              [
                {
                  "min": "0",
                  "max": "7",
                  "meta": "Betriebsart",
```

```
  "options":
  [
    "Stop",
    "Konstantspg.",
    "Mpp-Betrieb",
    "Res1",
    "Res2",
    "Res3",
    "Res4",
    "Res5"
  ],
  "value": "Mpp-Betrieb",
  "name": "Betriebsart",
  "unit": ""
},
{
  "min": "2150",
  "max": "2150",
  "meta": "Plimit",
  "value": "2150",
  "name": "Plimit",
  "unit": "W"
},
{
  "min": "0",
  "max": "4294900000",
  "meta": "SMA-SN",
  "value": "2000101000",
  "name": "SMA-SN",
  "unit": ""
},
{
  "min": "125",
  "max": "600",
  "meta": "Upv-Start",
  "value": "150",
  "name": "Upv-Start",
```

```
  "unit": "V"
 },
 {
  "min": "1",
  "max": "300",
  "meta": "T-Stop",
  "value": "4",
  "name": "T-Stop",
  "unit": "s"
 },
 {
  "min": "125",
  "max": "600",
  "meta": "Usoll-Konst",
  "value": "600",
  "name": "Usoll-Konst",
  "unit": "V"
 },
 {
  "min": "5",
  "max": "300",
  "meta": "T-Start",
  "value": "10",
  "name": "T-Start",
  "unit": "s"
 },
 {
  "min": "0",
  "max": "100",
  "meta": "Software-SRR",
  "value": "2",
  "name": "Software-SRR",
  "unit": "Version"
 },
 {
  "min": "0.005",
  "max": "4",
```

```
  "meta": "dFac-Max",
  "value": "0",
  "name": "dFac-Max",
  "unit": "Hz/s"
 },
 {
  "min": "0",
  "max": "7",
  "meta": "Storage",
  "options":
  [
   "permanent",
   "volatile",
   "Res1",
   "Res2",
   "Res3",
   "Res4",
   "Res5",
   "Res6"
  ],
  "value": "permanent",
  "name": "Storage",
  "unit": ""
 },
 {
  "min": "0",
  "max": "100",
  "meta": "Software-BFR",
  "value": "2",
  "name": "Software-BFR",
  "unit": "Version"
 },
 {
  "min": "0",
  "max": "100",
  "meta": "Hardware-BFS",
  "value": "1",
```

```
            "name": "Hardware-BFS",
            "unit": "Version"
          }
        ]
      }
    ]
  }
}
```

7.7 RPC_SET_PARAMETER

7.7.1 Version 1.0

Sets parameter values of up to 5 devices and returns the device list submitted in the request with the corresponding current parameter values. A check to see whether every parameter value was set successfully must be carried out by the application used.

Structure:

```
RPC={
    "version": "1.0",
    "proc": "SetParameter",
    "id": "ID",
    "format": "FORMAT",
    "passwd": "PASSWORT",
    "params":
    {
      "DEVICES":
      [
        {
          key,
          "channels": [CHANNELS]
        }
      ]
    }
}
```

A list with the device objects whose parameters are to be changed must be transferred as a parameter. Every device object contains a list with the parameters that must be set.

Configuration of the parameters is performed synchronously. Thus, the response time depends on the number of parameters to be set. For the next example, the response time is approx. 10 seconds.

PASSWORD: The MD5 coded hash value of the password for the desired access level.

DEVICES: An array containing objects with the device keys of the devices, and an array with the CHANNELS whose parameter values are to be set.

DEVICE_KEY: The corresponding device key (see Section 6.1 "Device Object", page 17).

CHANNELS: An array containing the channel objects to be set for the respective device. A list with the available parameter channels can be obtained by means of the command RPC_GET_PARAMETER_CHANNELS.

Sample request:

```
RPC={
    "version": "1.0",
    "proc": "SetParameter",
    "id": "1",
    "format": "JSON",
    "passwd": "a289fa4252ed5af8e3e9f9bee545c172",
    "params":
    {
     "devices":
     [
      {
       "key": "WR21TL06:2000101000",
       "channels":
       [
        {
         "meta": "Betriebsart",
         "value": "Mpp-Betrieb"
        }
       ]
      },
      {
       "key": "WR21TL06:2000101001",
       "channels":
```

```
            [
              {
                "meta": "Betriebsart",
                "value": "Stop"
              }
            ]
          }
        ]
      }
    }
```


Sample response:


```
RPC={
        "version": "1.0",
        "id": "1",
        "format": "JSON",
        "proc": "SetParameter",
        "result":
        {
          "devices":
          [
            {
              "key": "WR21TL06:2000101000",
              "channels":
              [
                {
                  "min": "0",
                  "max": "7",
                  "meta": "Betriebsart",
                  "options":
                  [
                    "Stop",
                    "Konstantspg.",
                    "Mpp-Betrieb",
                    "Res1",
                    "Res2",
                    "Res3",
```

```
            "Res4",
            "Res5"
          ],
          "value": "Mpp-Betrieb",
          "name": "Betriebsart",
          "unit": ""
        }
      ]
    },
    {
      "key": "WR21TL06:2000101001",
      "channels":
      [
        {
          "min": "0",
          "max": "7",
          "meta": "Betriebsart",
          "options":
          [
            "Stop",
            "Konstantspg.",
            "Mpp-Betrieb",
            "Res1",
            "Res2",
            "Res3",
            "Res4",
            "Res5"
          ],
          "value": "Stop",
          "name": "Betriebsart",
          "unit": ""
        }
      ]
    }
  ]
 }
}
```

8 Compliance Information

FCC Compliance

This device complies with Part 15 of the FCC Rules. Operation is subject to the following conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

The user is cautioned that changes or modifications not expressly approved by SMA America, LLC could void the user's authority to operate this equipment.

IC Compliance

The Sunny WebBox with *Bluetooth* complies with Industry of Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions:

- This device may not cause interference, and
- This device must accept any interference, including interferences that may cause undesired operation of the device.

The Sunny WebBox complies with Canadian ICES-003.

RF Exposure

The Sunny WebBox with *Bluetooth* should be installed and operated with minimum distance of 8 in. (20 cm) between the radiator and your body.

9 Contact

If you have technical problems concerning our products, contact the SMA Service Line.
We require the following information in order to provide you with the necessary assistance:

- Operating system of your computer
- Sunny WebBox firmware version
- Sunny WebBox serial number and hardware version
- Type of communication interface between Sunny WebBox and the inverters
- Type and serial numbers of the inverters connected to the plant

Country	Company Name	Headquarters	SMA Service Line
America	SMA America, LLC	Rocklin, CA	+1 877-MY-SMATech (+1 877-697-6283)* +1 916 625-0870**
Canada	SMA Canada, Inc.	Toronto	+1 877-MY-SMATech (+1 877-697-6283)***

* toll free for USA, Canada and Puerto Rico
** international
*** toll free for Canada

SMA Solar Technology

www.SMA-Solar.com

SMA America, LLC
www.SMA-America.com